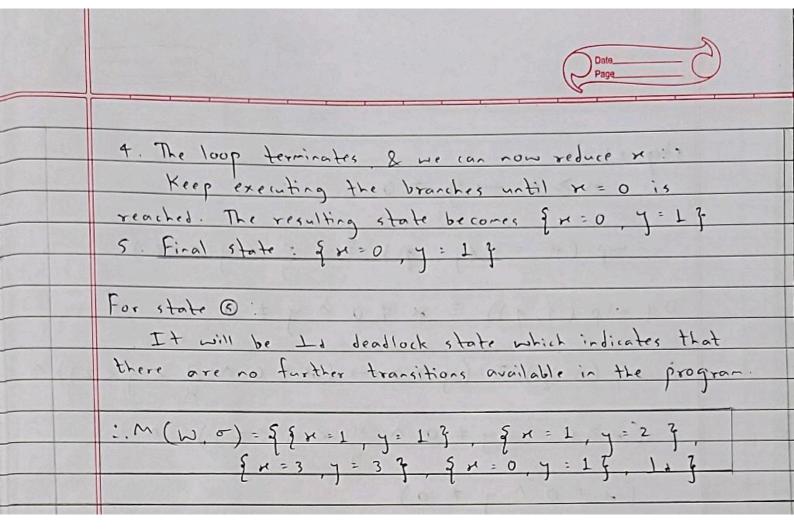


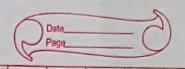
For state 1: 1. starting from [x=3, y=1 } Branch 2: x 7y -> y := y +1

condition: 3 > 1 is true -. y:= 1+1=2 => New state: { x=3, y=2} 2. from { x = 3 , y = 2 }: Branch 2: x > y -> y := y + 1 condition: 3 > 2 is true : y:= 2+1 =3 => New state: {x=3, y=3} 3. Final state: {x=3, y=3} for state @: 1. Starting from & x = 3 , 7 = 1 7 : Branch 1: x > y -> x := x - 1 condition: 3>1 is true : x:=3-1=2 => New state: [x=2, y=1] 2. From { x = 2 , y = 1 } ! Branch 1: x7y -> x:=x-1 condition: 271 is true : N:= 2-1 = 1 => New state: { x=1 , y=1 } 3. From & x=1, y=1 3. Branch 1: x77 -> x:=x-1 condition: 1 > 1 is talse Branch 2: x>y -> y := y + 1 condition: 1 > 1 is false Branch 3: x+y = 4 -> x := y/x condition: 1+1=2 i.e. false because its not + Branch 4: x+y = 4 -> x:= x/y condition: i+1=2 i.e. talse because its not +

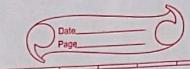


	10/19	active set one experience different to the settlement
	200	for her to work he wish & proposer with the
Q.	2 .	defining the so hand not of it is a fall and a
	<u> </u>	MAJORITY = $k_0 = 0$; $k_1 = 0$;
		while ko <n <="" do="" j;<="" k1="" n="" td="" x=""></n>
		Ko:= ko+1;
		$k_1 := k_1 + 1 \text{ od}$
2		if $k_0 = n \longrightarrow major := 21 \square k_1 = n \longrightarrow major := 0 fi$
		A DE LOCAL DE LA PORTE DE LA P
		Here J = do b[ko] = 1 -> ko := ko + 1 D b[k1] = D ->
		$k_1 := k_1 + 1 \text{ od}$
	de	refrance of all a little and a make was later a
		full allowing all a stables what que question
		estat a will all almost a rough the

a.	3.	
	0)	If M(s, or) contains exactly one state, then 5 must
		If M(s or) contains exactly one state, then 5 must be a deterministic statement.
	\rightarrow	Tone
		Explanation. If the result of applying m (5 , or) results in exactly one state it indicates that there is no
		in exactly one state it indicates that there is no
		nondeterminism in the execution of s. Because of the
		key property of deterministic systems where for every
		input, the system has only one possible next state.
	bJ	If o = Sp3 S (qf, then o = p.
	->	talse
		Explanation: If o sutisties the triplet, then or may
		or may not satisfies p. And we know that
		if of # p then o = {p} 5 {q}
		If o \$ tot [p] 5 {q}, then o \$ p
	->	false
		Explanation: If of the gpg s gq t this does not typically mean or imply that of the Because the failure to prove q from p could occur for various reasons. Logically if ofton Spg s gq then of p n M (s, o) to q
		typically mean or imply occur for various reasons.
		to prove of if offer Spassay then of prom (50) # 9
		logically.
-	1	



d) If = = {p} 3 5 {q}, then m (5,0) = 9 Explanation: If of FSpg SSqg & or is partially correct then M(S, o)-1 Fq. Therefore, M(5,0) Fq is incorrect & may contain prevdo e) If offp 3 s {q3, then of the 1p3 s {23 -> True Explanation of # {p35 [9] : If partial correctness does not hold it means that the program may not lead to the correct postcondition q even if it terminates. of # tot &p 1.5 [q4: Since total correctness includes partial correctness as part of the definition, it follows that total correctness also does not hold in this case so, if of Epgs 199 then of #tot 2pgs 199 is correct. a) 3P(k, s+1) } s := s+1 {P(k, s) } Explanation: Applying the backward assignment sale will satisfy the triplet Explanation: Let o = {5=1, k=1 } . We have = FP(1,1), but m (s=s+1, 0) = {s=2, k=1} # p(1,2)

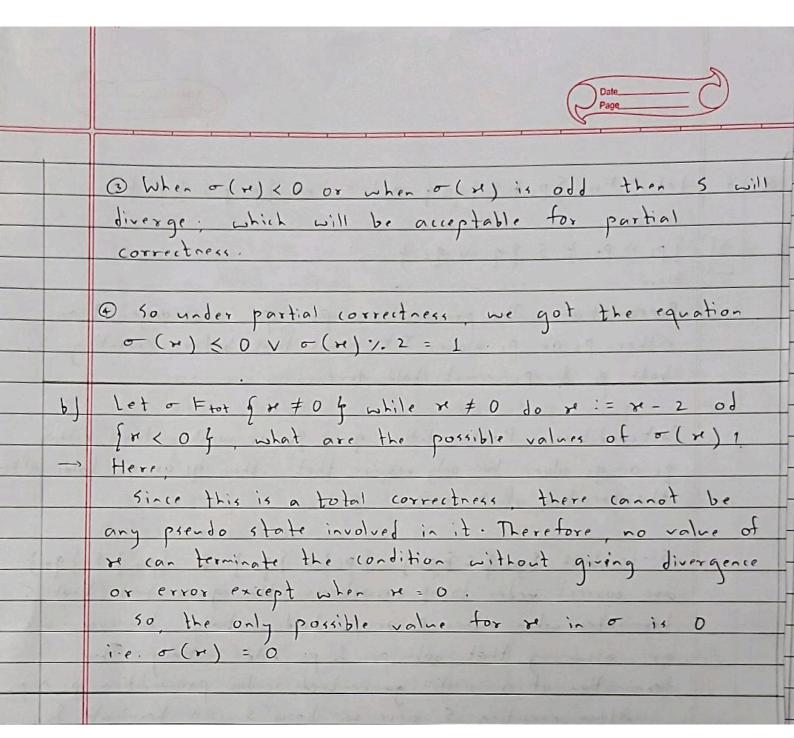


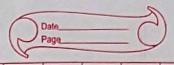
e) {P(k,s) 1, s < 0 4 s := s + 1; k := k + 1 {P(k,s)} Explanation: The above statement is True because the precondition is a contradiction & cannot be satisfied by any state. d) {P(k,s) \s = 50 \forall s := s + 1 {P(k, 50) } Explanation: The precondition logically implies P(k,s) A
P(k, so): Even after incrementing s, the post
condition does not change so P(k,so) is still true. e) {P(k+1, s+1)} s:=s+1; k := k+1 {P(k, s)} Explanation: Using backward substitution rule, we get {P(k+1, s)} k!= k+1 {P(k,s)} and {p(k+1,5+1)} + 5:= 5+1 { k(k+1), 5)} Combine them using the sequence rule & we get: {P(k+1,5+1)} 5:=5+1, k:= k+1 {P(k,5)}. Let $\sigma \models \{x \neq 0 \}$ while $x \neq 0$ do x := x - z od $\{x < 0\}$.

what are the possible values of $\sigma(x)$?

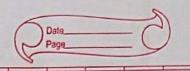
Here, possible values of o (x): o when $\sigma(r) = 0$, then the precondition is not satisfied so the triple is satisfied @ When o(4)70 and o(4) is even, 5 will terminate with some state T with T (r) = 0, which ultimately

does not satisfy the post condition





Q. 6. Let = Ftot { p. 4 5 { 9 1 4 and = Ftot { p. 3 5 { 9 2 4 1 p. 1 p. 3 5 9 91 V 92 9 Precondition: The precondition pr 1 pz is stronger than either pror pralone. It means that before executing s both pr & pr must hold true. Postcondition: Postcondition qu v 92 is weaker than either or or of alone. We only require that either of or of holds after the execution of s. Since both {pr } & {pr} & {pr are correct under total correctness this quarantees that s terminater when either Pr or Pr holds initially however we are assuming that both pr & pr hold initially. Thus termination of 5 is still guaranteed under precondition pr 1 p. After executing 5, since we know 5 will terminate & either 9s or 92 will be true based on initial correctness conditions of p. & pz we conclude that q, v qz will satisfied -. IPINP2 3 5 Eq. v 92 3 is true under total correctness. b) { p1 v p2 4 5 { 91 x 92 5 Since { p1 & 5 & q1 & and } p2 & 5 & 92 & are correct under total correctness, we know that 5 will terminate when either prox pr holds initially. This guarantees that s will always terminate when P1 v P2 holds.



If the precondition is p. vp. then:

i) If p. holds initially, we know that S will terminate &

q. will hold, but we cannot guarantee that q. will hold.

ii) If p. holds initially, we know that S will terminate

& q. will hold, but we cannot guarantee that q. will

hold.

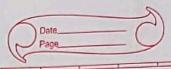
Therefore, we cannot guarantee that both 91 & qe will hold after execution it only one of pror pris true initially.

ingpr v pr & S gar A gr & does not hold true under total correctness.

· | { p1 v p2 } 5 { q1 v p 92 }

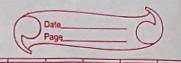
Since Spifs Sqif and Ep. & Sqif are correct under total correctness. Swill terminate if either Pi or Pi holds initially. Thus termination of S is guaranteed under the precondition Pi V Pi If p. holds initially, then by the total correctness of [Pi i S Sqi i S will terminate and qi will hold which satisfies qi v qi. If pi holds initially, then by the total correctness of [Pi i S Sqi i S will terminate and qi will hold which satisfies qi v qi. If pi holds initially, then by the total correctness of [Pi i S Sqi i S will terminate and qi will hold, which also satisfies qi v qi. will be satisfied.

correctness.



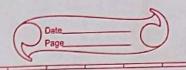
Q. 7 let = {p, 35 {q, 3 and = {p, 35 {92}} a) fp1 ^ p2 f S { 91 ^ 92 f Since & p. 3 S & 913 holds, whenever P1 is true before executing 5 , qu will hold afterward. Similarly since [pz] S {qz} holds, whenever

pz is true befor executing S, qz will hold after-Given that both pr & pr are true initially because P1 A p2 & since 's guaranters that q2 follows from
P1 & p2 q2 follows from P2, the post condition 91 A 92 will also hold. -. { P1 1 P2 } 5 } 91 1 92 } is valid under the partial correctness. b) { P2 3 5 8 91 -> 92 5 Since, pr is true before execution, according to the given assumption & p. & S goz & ge will be true after executing s. For Implication 91 -> 92: It qu'is true after executing 5, since we know of is also true after executing 5. the implication 9. -> q. holds true. If go is talse after executing s, the implication 91 -> 92 also holds because a false antecedent makes the implication true regardless of the truth value of 92.

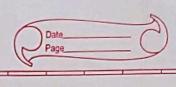


:. Since in both scenarios the implication q1 - , 92 holds, we can conclude that gp= 3 5 8 91 -> 92} is valid under partial correctness () { ¬ p1 → p2 3 5 { ¬q1 → q2 } to evaluate the validity, we consider the implication of the precondition if apr is true by the precordition apr pr must be true before executing 5. Since p2 is true by the assumption [pz] 5 \ 92 \ we can conclude that 92 will be true after executing 5. Considering of the truth value of go is not quaranteed trom the precordition Tp. -> pz. If p. is tree before executing & then next is implication: Implication of postcondition -19, -> 92:

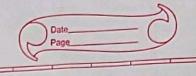
If 191 is true because executing 5 from our analysis. if age is true we need to verify that que is true. Pr is true then go is guaranteed to the be true tron the assumption 1p. 7 5 fgz 3. correctioness.



Q-8. a) Ftot Sw & 5 { 9 4 Explanation: The weakest precondition up (5, 9) is specifically designed to ensure that the program S, when executed from a state satisfying w will terminate & establish the postcondition of since the program is deterministic the total correctness will hold. b) = 5w > 9 5 5 9 4 Explanation: Here, wag is a stronger precondition than w because w is the weakest precondition, also if all states totally satisfy a triple, then they also partially satisfy the triple. M (s, o) Fq Explanation: If o I w , there is no guarantee that executing 5 from or will yield q. The definition of the weakest precondition wis meant to ensure that if it Soesn't hold we cannot assert anything about the validity of g after execution 11 If of w, then M (5,0) \$ \$13. - True Explanation: or = w means program S execute without undefined states. And M(s, o) # 1 is true as long as o is well formed & satisfies w. so the above statement is True.



e) It o 1 + w, then o = g - w 3 5 5 - 93 Explanation: If o 1 w, it does not tollow that o = [nw } s { ng } is true. Just because the precondition does not hold does not guarantee that the postcondition be determined solely based on the failure of the weakest precondition. Q. 9. let 5 = y := y % x and g = sgrt (y) > x. Calculate wip (5, 9) 5 = y := y 1/2 x and g = sgrt (g) > x The wip rule for an assignment statement y := E is w/p (y:= E, q) = 9 [y/E] where, gly/E] means "substitute E for y in 9" In this case, E is (y 1. re) so, we need to substitute (9 %. x) for y in q :. w/p (y:=y ./. x, sqrt (y) > x) = sqrt (y 1. x) > x1 Therefore the final result is: w/p(5, 9) = sqrt (y 1. x) > x Calculate wp (5, 9): S = y := y : x and q = sqrt (y) > x wp (5, 9) for loop free programs:



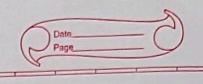
wp (5, 9) = D(5) 1 whp (5, 9) 1 D (whp (5, 9)) $D(5) \equiv D(y:=y'). \Re) \equiv \chi \neq 0$ w/p (5, q) = sqrt (y -/. x) > xe D(w/p(5,q)) = D (5qrt (y 1/2 x) > xe) = y 1/2 x 2 0 x x \$ 0 :. wp (5, q) = (x + 0) ^ (sqrt (y ./. x)) > x ^ Q. 10. Let 5 = if y >, 0 -> x := y /x D x >, 0 -> x := x/y

fi and q = x < y < z. a) calculate w/p(5, 9) for the weakest liberal precondition (wlp) we want the weakest condition that must hold before exactly sexecuting 5 such that if 5 terminates q holds .. if for branch y > 0 the postcondition becomes 7 < 4 < 2

because after n:= y/m, re is replaced by y/m
in postcondition.

iif For branch 170 the postcondition becomes

	Because after x:= x , x is replaced by x
	9 / 9
	in the post condition.
	:. w/p (5,q) = (y >, 0 -> = +y/x < y < 2) 1
	(x7,0 -> x/y < y < 2)
61	(alculate wp (5, q)
→	
	D(5) = (420 -> x + 0) \ (x20 -> y + 0)
	step ij:
	77,0 -> x + 0
	3
	:. wp (x:=y/n), q) = (y < y < z)
	a rate itself



step 2:

:. wp (x:=x/y; q) = (x < y < z)

D(w/p(5,9)) = x + 0 1 y + 0

:. wp (5, q) = (y > 0 -> (x + 0 ^ y < y < z)) ^

 $(x \ge 0 \rightarrow (y \ne 0 \land x \land y \land z)) \land$

(x # 0 / y # 0)

18 18 -- 0 (P) = (, s) als

(p 12) on and the